

Brute-Force Hitori Solver

Everaldo Sembiring(13510095)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13510095@students.stei.itb.ac.id

ABSTRAK

Hitori adalah salah satu permainan teka-teki dari Jepang. Permainan ini menggunakan papan permainan sebesar $n \times n$, dimana pada papan tersebut sudah terisi penuh dengan susunan angka. Untuk menyelesaikan puzzle ini, maka pemain harus menghapus beberapa angka sehingga tidak terdapat angka yang sama dalam satu kolom dan row yang sama. Namun terdapat aturan lain untuk menghapus angka-angka tersebut.

Makalah ini akan menjelaskan aturan lengkap dari permainan tersebut. Begitu juga salah satu teknik penyelesaian permainan ini dengan menggunakan algoritma brute force. Dimana dapat menunjukkan langkah pasti untuk dapat menemukan solusi.

Kata kunci : hitori, brute-force.

I. PENDAHULUAN

Game puzzle adalah permainan yang berisi sebuah persoalan, dimana pemain bertugas untuk mencari solusi dari persoalan tersebut. Selain dapat berguna untuk mengisi waktu luang, game ini sangat berguna dalam meningkatkan daya pikir kita. Sehingga biasanya game puzzle dapat digunakan sebagai bahan acuan secepat apa kemampuan kita untuk menyelesaikan suatu permasalahan.

Salah satu negara yang menghasilkan game puzzle adalah negara Jepang. Di Jepang terdapat sebuah publisher yang sudah sering mempopulerkan game puzzle ke dunia. Nama dari publisher tersebut adalah Nikoli. Contoh game puzzle seperti Sudoku, Nikoli, dan teka-teki silang merupakan game puzzle yang dipopulerkan oleh Nikoli. Hitori adalah salah satu game yang dipopulerkan oleh Publisher Nikoli.

Nikoli adalah permainan yang menggunakan papan yang berisi angka yang mirip dengan Sudoku. Namun pada papan Nikoli, setiap grid sudah penuh terisi angka. Tugas dari pemain adalah menghapus angka sehingga tidak ada angka yang sama pada setiap baris dan kolom.

Untuk menyelesaikan permainan ini, sangat tidak efisien jika menggunakan metode trial and error. Akan sangat sulit untuk diselesaikan jika ukuran papan sudah sangat besar. Maka salah satu metode pasti untuk menyelesaikan permasalahan ini adalah algoritma brute force.

1.1. Aturan Permainan Hitori

Tujuan utama dari permainan hitori adalah memberi warna hitam pada beberapa angka pada papan, dengan kriteria sebagai berikut:

1. Ketika selesai member warna hitam, tidak boleh ada kotak putih yang memiliki nilai yang sama pada row dan kolom dari kotak tersebut.
2. Tidak boleh ada 2 warna hitam yang saling bersebelahan.
3. Setiap kotak putih harus terhubung.

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 5 | 4 | 1 | 2 | 3 |

Gambar 1: Contoh Puzzle Hitori
(<http://www.cs.rit.edu/~zjb/courses/ai/proj1.html>)

Pada Contoh diatas, pada kolom kedua terdapat 3 angka 3. Menurut aturan pertama, Salah satu dari ketiga angka tersebut akan ada bewarna putih, dan sisanya adalah hitam. Bilang 3 ditengah diberi warna hitam, Maka sisanya tidak dapat diberi warna hitam karena melanggar aturan ke-2. Jadi angka 3 ditengah dibiarkan bewarna putih kemudian menghitamkan angka 3 di atas dan di bawah. Kemudian ada 3 angka 4 di row ketiga. Jika dipaling kiri dibiarkan bewarna putih, sisanya harus diberi warna hitam, dan melanggar aturan ke-2. Jadi yang dipaling kiri harus diberi warna hitam. Sehingga kotak tersebut menjadi:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 5 | 4 | 1 | 2 | 3 |

Gambar 2

(<http://www.cs.rit.edu/~zjb/courses/ai/proj1.html>)

Dari gambar diatas,nilai 2 yang ada ditengah tidak bisa diberi warna hitam,karena dapat mengisolasi angka 3 disebelahnya.Hal tersebut melanggar aturan ke-3.Kemudian terdapat 2 angka 5 dibagian atas.angka 5 pada kolom ke 3 juga tidak dapat diberi warna hitam,karena dapat mengisolasi angka disebelahnya.Maka yang diberi warna hitam pada angka 5 di bagian paling kanan.Demikian seterusnya sehingga solusi yang terbentuk adalah sebagai berikut:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 5 | 4 | 1 | 2 | 3 |

Gambar 3

(<http://www.cs.rit.edu/~zjb/courses/ai/proj1.html>)

1.2. Pengertian Brute Force

Brute force adalah pendekatan yang lempang untuk memecahkan suatu masalah.Biasanya didasarkan pada:

- pernyataan masalah (*problem statement*)
- definisi konsep yang dilibatkan.

Algoritma *brute force* memecahkan masalah dengan,sangat sederhana,,langsung, jelas (*obvious way*).

II. METODE

2.1. Gambaran Solusi

Untuk menyelesaikan permasalahan puzzle hitori menggunakan brute force,maka akan dilakukan iterasi satu persatu setiap kotak dari papan.Di setiap kotak,akan dilakukan dengan perbandingan dengan pattern-pattern dibawah ini.Jika ada yang sama maka akan dilakukan aksi terhadap kotak tersebut.Berikut adalah pattern-pattern yang akan dipakai:

Pattern 1

| | | |
|---|---|---|
| 6 | 6 | 6 |
| 4 | 5 | 9 |
| 1 | 2 | 3 |

| | | |
|---|---|---|
| 6 | 6 | 6 |
| 4 | 5 | 9 |
| 1 | 2 | 3 |

Penjelasan:

Program dimulai dari kiri atas.Program akan mengecek apakah terdapat angka yang sama pada satu row.Ternyata ditemukan angka 6 yang lain.Program akan menghitamkan kotak yang kiri atas.Setelah menghitamkan program akan menandai bahwa angka 6 pada baris kedua dan angka 4 pada row kedua pasti berwarna putih.Karena tidak mungkin terdapat warna hitam berdampingan.Karena angka 6 pada kolom kedua pasti putih,kemudian dilanjutkan pada angka 6 yang ketiga.Pada saat pengecekan 1 row,ditemukan bahwa terdapat angka 6 yang sudah pasti berwarna putih,maka pastilah kotak tersebut berwarna hitam dan angka 9 sudah pasti berwarna putih.

Pattern 2

| | | |
|---|---|---|
| 6 | 3 | 2 |
| 6 | 1 | 9 |
| 6 | 2 | 3 |

| | | |
|---|---|---|
| 6 | 3 | 2 |
| 6 | 1 | 9 |
| 6 | 2 | 3 |

Sama seperti pada pattern pertama,namun dilakukan pengecekan pada 1 kolom.Sehingga yang berwarna hitam adalah 6 pada row 1 dan 3.Kemudian yang sudah pasti berwarna putih adalah 6 ditegah,3 pada row 1,dan 2 pada row 3.

Pattern 3

| | | |
|---|---|---|
| 6 | 6 | 2 |
| 4 | 6 | 9 |
| 1 | 2 | 3 |

| | | |
|---|---|---|
| 6 | 6 | 2 |
| 4 | 6 | 9 |
| 1 | 2 | 3 |

Pada pattern ketiga ini, setiap kotak dibandingkan pada row dan kolom. Program dimulai dari 6 kiri atas. Ternyata pada kotak tersebut hanya memiliki angka yang sama pada row saja. Sehingga tidak dilakukan apa-apa pada kotak tersebut. Kemudian selanjutnya mengecek 6 pada row 1 kolom kedua. Ternyata terdapat angka yang sama pada row 1 dan kolom 2. Sehingga kotak tersebut diberi warna hitam. Kemudian angka 6 pada kolom 1, angka 6 pada row 2, angka 2 pada row 1 sudah pasti berwarna putih.

Pada implementasi program, pertama kali seluruh kotak akan dibandingkan dengan pattern 3 lebih dahulu. Kemudian dibandingkan kembali dengan pattern 1 dan 2.

2.2. Implementasi Algoritma Dengan Brute Force

Pseudo-code algoritma brute force untuk penyelesaian Hitori adalah sebagai berikut:

1. Deklarasi

{Kamus Global}

```
const N : integer = 10
type MatrikInteger : array[1..N, 1..N] of integer
type MatrikChar : array[1..N][1..N] of integer
```

```
intBoard : MatrikInteger
solvedBoard : MatrikChar
```

- Asumsi bahwa panjang board yaitu N, sebanyak 10.
- MatrikInteger nantinya adalah isi dari papan board yang sebenarnya. Matrik adalah yang akan ditelusuri, namun pewarnaan kotak bukan di matrik ini.
- MatrikChar adalah matrik yang merupakan menjadi solusi dari permasalahan. Matrik ini berisi warna dari setiap kotak. Matrik ini nantinya akan merepresentasikan kotak mana yang berwarna hitam dan kotak mana yang berwarna putih.

2. Prosedur mengisi nilai awal dari solvedBoard

```
procedure newboard(output
solvedBoard:MatrikChar , input
intBoard:MatrikInteger)
```

N : integer

N <= panjangpapan(intBoard)

For row <= 1 to N do

For kolom <= 1 to N do

solvedBoard_{row,kolom} = '?'

endfor

endfor

Pada bagian ini adalah inialisasi nilai dari solvedBoard sebelum mencari solusi. Fungsi panjangpapan adalah mengembalikan panjang papan permainan yang sebenarnya. Pada saat deklarasi, panjang matriks adalah sebanyak N. Namun ada kemungkinan yang diisi < N.

Pada solved board, setiap kotak diberi karakter '?'. Huruf '?' merupakan penanda bahwa kotak tersebut belum tau apakah berwarna hitam atau putih.

3. Prosedur membandingkan dengan pattern 3

```
procedure solve3(output solvedBoard:MatrikChar ,
input intBoard:MatrikInteger)
```

N : integer

sum : integer

N <= panjangpapan(intBoard)

For row <= 1 to N do

For kolom <= 1 to N do

if(solvedBoard_{row,kolom} = 'W' or
solvedBoard_{row,kolom} = 'B')

nextkolom()

endif

sum <= 0

if(checkrow(row, kolom,
intBoard_{row,kolom}) = false)

sum++

endif

if(checkkolom(row, kolom,
intBoard_{row,kolom}) = false)

sum++

endif

if(sum = 2)

solvedBoard_{row,kolom} = 'B'

solvedBoard_{row+1,kolom} = 'W'

```

        solvedBoardrow-1,kolom='W'
        solvedBoardrow,kolom+1='W'
        solvedBoardrow,kolom-1='W'
    endif

    else if (sum=0)
        solvedBoardrow,kolom='W'
    endif

endfor
endif

```

Prosedur ini akan menelusuri seluruh tabel intBoard secara satu persatu. Jika pada row dan kolom tertentu dan ternyata pada charBoard sudah tertanda 'B' atau 'W', maka kotak tersebut tidak perlu diperiksa dan langsung lanjut ke kotak selanjutnya. Kemudian kotak akan diperiksa berdasarkan row dan kolomnya. Sum berfungsi sebagai penanda, jika 1 maka ada value yang sama pada row atau kolom, bukan keduanya. Jika 2 maka terdapat value yang sama pada row dan kolom tersebut. Jika sum=2 maka kotak pada row dan kolom tersebut diberi warna hitam. Kemudian memberikan warna putih pada bagian atas, kanan, bawah, kiri dari kotak tersebut. Jika sum=1 artinya kotak tersebut memiliki kemiripan dengan pattern 1 atau 2, namun pada prosedur ini hanya melakukan perbandingan dengan pattern 3. Sehingga tidak dilakukan apa-apa. Jika pattern=0 artinya tidak ada yang sama pada baris dan kolomnya. Sehingga diberi warna putih. Setelah itu melakukan iterasi kotak berikutnya dan mengulangi pengecekan kotak lagi. Hingga iterasi seluruh kotak tercapai.

4. Prosedur membandingkan dengan pattern 1 & 2

```

procedure solve1and2(output solvedBoard:Matrikchar ,
input intBoard:MatrikInteger)

```

N : integer

sum : integer

N <= panjangpapan(intBoard)

For row <= 1 to N do

For kolom <= 1 to N do

```

        if(solvedBoardrow,kolom='W' or
        solvedBoardrow,kolom='B')
            nextkolom()
        endif

```

endif

```

        if(checkrow(row,kolom,
        intBoardrow,kolom)=false

```

or

```

        checkkolom(row,kolom,
        intBoardrow,kolom)=false)

```

```

            solvedBoardrow,kolom='B'

```

```

            solvedBoardrow+1,kolom='W'

```

```

            solvedBoardrow-1,kolom='W'

```

```

        solvedBoardrow,kolom+1='W'
        solvedBoardrow,kolom-1='W'
    endif

    else
        solvedBoardrow,kolom='W'
    endif

endfor
endif

```

Prosedur ini mirip dengan prosedur solve3. Perbedaannya adalah hanya pada perbandingan row dan kolom. Jika salah satu saja dari row dan kolom memiliki nilai yang sama, maka kotak tersebut langsung diberi warna hitam. Bila tidak ada yang sama maka akan diberi warna putih.

5. Fungsi mengembalikan solusi akhir

```

function solver(output solvedBoard:Matrikchar ,
input intBoard:MatrikInteger)=>Matrikchar

```

```

    solve3(solvedBoard,intBoard)

```

```

    solve1and2(solvedBoard,intBoard)

```

```

    return solvedBoard

```

Fungsi ini akan menjalankan prosedur solve 3 terlebih dahulu. Sehingga semua kotak yang memiliki nilai yang sama pada row dan kolomnya akan diberi warna hitam. Pada prosedur ini juga sudah terbentuk sebagian kotak berwarna putih.

Kemudian dilanjutkan dengan fungsi solve1and2, sehingga sudah pasti semua kotak berwarna putih atau hitam. Hasil akhirnya prosedur tersebut kemudian dikembalikan oleh fungsi.

III. Hasil Analisis

3.1. Hasil

Dengan menggunakan algoritma ini, ternyata dapat menyelesaikan puzzle hitori yang sederhana, contoh persoalan untuk dianalisis:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 5 | 4 | 1 | 2 | 3 |

Pada saat menjalankan fungsi solve3, solverBoard akan berisi sebagai berikut:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 5 | 4 | 1 | 2 | 3 |

Dapat dilihat dari gambar diatas jika semua value yang memiliki nilai yang sama pada row dan kolomnya. Pasti bewarna hitam dan yang dilingkari adalah kotak yang sudah pasti bewarna putih. Sedangkan yang belum dilingkari adalah kotak yang sebelumnya dicek dan ternyata pada row atau kolomnya ada yang sama, namun bukan keduanya. Sehingga dibiarkan kosong.

Kemudian program melanjutkan prosedur solve1 and 2:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 5 | 4 | 1 | 2 | 3 |

Dari hasil tersebut ternyata algoritma brute force ini dapat menyelesaikan persoalan yang sederhana.

Karena seluruh kotak diiterasi sebanyak 2 kali, maka kompleksitas algoritmanya adalah sebanyak $O(n)$.

3.2. Kekurangan

Ternyata algoritma brute force ini tidak berhasil melengkapi aturan yang ketiga. Contoh kasus kesalahan yang terjadi adalah sebagai berikut:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 2 | 4 | 1 | 2 | 3 |

Solusi akhir yang dicapai:

| | | | | |
|---|---|---|---|---|
| 3 | 2 | 5 | 4 | 5 |
| 2 | 3 | 4 | 3 | 5 |
| 4 | 3 | 2 | 4 | 4 |
| 1 | 3 | 3 | 5 | 5 |
| 2 | 4 | 1 | 2 | 3 |

Hal ini terjadi karena pada saat menjalankan prosedur solve3. Pada posisi (5,1), program memperoleh bahwa terdapat angka yang sama pada posisi (2,1) dan (5,4). Sehingga angka tersebut diberi warna hitam. Akibatnya mengisolasi angka 1. Hal ini melanggar aturan ke 3, yaitu warna putih harus saling berhubungan.

Ternyata algoritma brute force kurang sempurna jika diimplementasikan pada permainan Hitori. Diperlukan heuristik brute-force yang lain untuk memecahkan solusi puzzle ini.

IV. Kesimpulan

Dari hasil analisis yang diperoleh ternyata algoritma brute force ini masih kurang sempurna untuk diimplementasikan pada permainan Hitori Solver karena belum dapat mematuhi aturan yang ketiga. Namun algoritma ini dapat memenuhi aturan 1 dan 2.

REFERENCES

- [1] <http://forums.xkcd.com/viewtopic.php?f=3&t=93478>
Waktu akses: 20 Desember 2012, 21.04 WIB
- [2] <http://www.cs.rit.edu/~zjb/courses/ai/proj1.html>
Waktu akses: 20 Desember 2012, 21.15 WIB
- [3] <http://www.cs.rit.edu/~zjb/courses/ai/proj1.html>
Waktu akses: 20 Desember 2012, 21.15 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 21 Desember 2012

ttd

A square image showing a handwritten signature in dark ink on a light-colored background. The signature is stylized and appears to be 'E. Sembiring'.

Everaldo Sembiring(13510095)